Subset Selection from Large Datasets for Kriging Modeling

Rennen, G.

Publication date:
2008

CentER

Discussion Paper

No. 2008–26

SUBSET SELECTION FROM LARGE DATASETS FOR KRIGING MODELING

By Gijs Rennen

March 2008

ISSN 0924-7815

Tilburg University

Subset selection from large datasets for Kriging modeling

Gijs Rennen

Department of Econometrics and Operations Research, Tilburg University,
P.O. Box 90153, 5000 LE Tilburg, The Netherlands

G.Rennen@uvt.nl

March 7, 2008

Abstract

When building a Kriging model, the general intuition is that using more data will always result in a better model. However, we show that when we have a large non-uniform dataset, using a uniform subset can have several advantages. Reducing the time necessary to fit the model, avoiding numerical inaccuracies and improving the robustness with respect to errors in the output data are some aspects which can be improved by using a uniform subset.

We furthermore describe several new and current methods for selecting a uniform subset. These methods are tested and compared on several artificial datasets and one real life dataset. The comparison shows how the selected subsets affect different aspects of the resulting Kriging model. As none of the subset selection methods performs best on all criteria, the best method to choose depends on how the different aspects are valued. The comparison made in this paper can be used to facilitate the user in making a good choice.

Keywords: Design of computer experiments, dispersion problem, Kriging model, large non-uniform datasets, radial basis functions, robustness, space filling, subset selection, uniformity.
JEL codes: C0, C90

1 Introduction

1.1 Motivation

Kriging is an interpolation technique which finds its roots in geostatistics. The method is named after Krige, a South-African mining engineer, and is based on his work at the Witwatersrand reef complex (Krige 1951). In the 1960s, the French mathematician Matheron formalized Krige's method (Matheron 1963). Besides geostatistics, Kriging has also found numerous applications in other fields. Sacks et al. (1989) applied Kriging in the field of deterministic simulation for the design and analysis of computer experiments. Since then many others followed; see Jones et al. (1998), Jones (2001), Koehler and Owen (1996), Santner et al. (2003), Stehouwer and Den Hertog (1999). A basic description of Kriging can be found in Appendix E.

When building a Kriging model, the general intuition is that using more data will always result in a better model. Therefore, a large dataset is regarded as a good starting point for building a model. However, when the dataset we can use is already given, there are situations where using only a subset has certain advantages. Especially when the large given dataset is non-uniformly distributed over the whole design space, problems can occur. In this paper, we analyze and test methods to select subsets in order to reduce these problems.

Large non-uniform datasets can occur in several situations. The first situation we can think of is when we have a set of legacy data (Srivastava et al. 2004). Legacy datasets contain results of experiments, simulations or measurements performed in the past. These results are stored for future use to avoid having to generate them again. This is especially useful if there are many results or if the expenses to obtain these results are high. As this data is not generated especially for making a global model, it is often not uniformly distributed over the whole design space. Another reason for non-uniformity of legacy datasets can be that they contain measurements of a system which cannot be fully controlled.

A second situation is when the data is the result of a sequential optimization method. These methods often generate more data points near potential optima than in other regions (Booker et al. 1999). If we want to use this data to fit a global metamodel, we thus have to take into account that it contains clusters of points.

Thirdly, non-uniform data also occurs if models are coupled (Husslage et al. 2003). We call two models coupled if the output of the first model is input for the second. If we want to construct a metamodel of both models, we can

use a space-filling design of experiments for the first model. It could be argued that we could also do this for the second model. In some cases however, it is better to use the output of the first model. Although the input of the first model is space-filling, its output is often not. When we want to construct a metamodel of the second model, we thus have to use a non-uniform dataset.

These are just a number of situations where we can come across large non-uniform datasets. Using these sets directly to build our model can impose problems which can often be resolved by using a uniform subset. We call a subset uniform if the input data of the data points in the subset are "evenly spread" over the entire design space. Whether or not using a uniform subset is better, depends not only on the dataset but also on the chosen modeling method.

Important reasons for using a subset instead of the complete dataset are the following:

- **Creating training and validation set**
  The most common reason for using only part of the data as training data is validation. Splitting a dataset into a training and a validation set is a well-know validation method and is often done randomly (Cherkassky and Mulier 1998, Golbraikh and Tropsha 2002). In the field of design of computer experiments (DoCE) however a consensus is reached that designs used for deterministic computer experiments should be space-filling (Simpson et al. 2001). A design is called space-filling if it fills the whole design space. For a given number of design points, the design space is best filled if the design points are evenly spread over the whole design space. Therefore, it would be a good idea to also take a uniform subset when selecting a training set from an existing dataset (Golbraikh et al. 2003).

- **Time savings**
  A second reason could be the reduction in time necessary to fit the Kriging model. This is certainly an important issue as its time-consumption is generally regarded as one of the main drawbacks of the Kriging method (Jin et al. 2001). With current implementations and computing power, it is sometimes even impossible to fit a Kriging model using all available data. For instance, when using the Matlab toolbox DACE provided by Lophaven et al. (2002) on a PC with a 2.4-GHz Pentium 4 processor, we encountered problems for datasets containing 3000 points or more.

  Especially the inversion of the correlation matrix can impose problems as this requires much time and memory capacity. Kriging is however not the only method which can benefit from using less data. Genetic programming could also significantly benefit as it requires a lot of models to be fitted to a training set (Koza 1992, Banzhaf et al. 1998). Often this forms a large part of the total computation time and is thus worthwhile to reduce.

  Besides fitting the Kriging model, also the prediction of new points requires less time because using less training data results in simpler models. This is mainly because the number of terms in the Kriging model depends on the size of the training set. The same holds for Lagrange interpolation. Especially in situations where the model is used for on-line monitoring and optimization, finding a fast and simple model is important (Kordon 2006).

- **Avoid numerical inaccuracies**
  A common property of large datasets is that they are non-uniform, which implies that they can contain points that lie very close together. This property can make the corresponding correlation matrix ill-conditioned (Davis and Morris 1997, Booker et al. 1999). Solving a linear system with an ill-conditioned matrix can cause significant numerical inaccuracies. The optimization of the Kriging parameters requires solving a linear system and can thus be inaccurate when data points lie close together. Removing certain points from the dataset can avoid the correlation matrix from becoming ill-conditioned and can thus improve the numerical accuracy of the Kriging model.

- **Improve robustness**
  Robustness with respect to errors in the output data can also be negatively influenced when data points lie close together. Siem and Den Hertog (2007) show that points that are close together can get assigned relatively large Kriging weights. Errors in these data points are thus magnified by the large Kriging weights. Removing some of the points can result in lower weights for the remaining points and thus a smaller effect of errors. We can thus sometimes improve robustness by using only a subset instead of the whole dataset.

These different motivations for subset selection require us to look at different performance criteria. In Section 4.2, we describe the criteria used in this paper to measure the effects of subset selection on these different factors.

For most of the mentioned motivations, it is important that the selected subset is uniform. Selecting a uniform set of points also occurs in other problems like Design of Computer Experiments (DoCE) and the dispersion problem. Although these are different problems, we can use some ideas from the fields of DoCE and dispersion problems in determining our subset. To see why and how we can use these ideas, we look at the similarities and differences between the problems.

1.2 Design of Computer Experiments

The most important reason why we can use ideas from DoCE is that it has the same aim as subset selection. In both problems the aim is to select the training set that will produce the model that most accurately approximates the function or process underlying the data. However, in practice we do not have an explicit description of the underlying function or process, which makes it impossible to directly optimize this objective. Instead we optimize certain properties of the training set that generally improve the quality of the resulting model. The most frequently used such property is space-fillingness (Simpson et al. 2001).

Another similarity is that in both problems we often have a restriction on the number of points we can use. In DoCE the restriction is caused by the large computation time per design point. The reasons for limiting the number of points by subset selection are given in Section 1.1.

Besides similarities, there are also differences between the two problems. The first difference is the set of points we can choose from. With DoCE, we can select all points in the design space. Sometimes we choose to restrict ourselves to points with a certain structure or property, like a Latin hypercube or an orthogonal array, but even then we are free in selecting which structure or property. With subset selection, we can only choose from the points in the original dataset. This implies, for instance, that the subset can only cover the design space as good as the original dataset does. It could be argued that sometimes additional points can be evaluated to improve for instance uniformity. We will however not take this option into account as it is beyond the scope of this paper.

The second difference is that the output values of all possible points are known in the case of subset selection. This in contrast to DoCE where we have to select our points without knowing their output values. Even in sequential DoCE, only the output values of previously selected and evaluated points are known. Using the output information in subset selection will most likely result in training sets which give more accurate models.

These differences show that we cannot directly use results from DoCE. In the paper by Srivastava et al. (2004) however, a method is described that uses a space-filling DoCE to determine a uniform subset. The main idea is to take a randomized orthogonal array and to select for each point of this orthogonal array, the data point closest to it. This idea can also be applied to other types of space-filling DoCEs. However, to limit the number of methods compared in this paper, we will only consider orthogonal arrays. In Section 3.1, the method of Srivastava et al. (2004) is described in more detail.

1.3 Dispersion problems

The dispersion problem can be described as follows. Given n possible locations, locate $m < n$ facilities such that some function of the distances between facilities is maximized (Ravi et al. 1994). Two commonly used functions are the minimum and the sum of the distances (Erkut and Neuman 1989). The first case is often referred to as MAXMIN and the second as MAXSUM. Both versions of the dispersion problem are strongly NP-hard. This was independently proven by Erkut (1990) and Ghosh (1996) for the MAXMIN problem and by Hansen and Moon (1994) and Kuo et al. (1993) for the MAXSUM problem. MAXMIN is also used as a measure of space-fillingness in DoCE. When we see locations as points, we can thus reformulate the problem as follows. Given n possible points, select a uniform subset of m points.

If we only focus on uniformity when selecting our subset and use MAXSUM or MAXMIN to measure the uniformity then subset selection is the same as the MAXSUM or MAXMIN dispersion problem. This however does not mean that we can directly apply all facility location algorithms and heuristics to the problem of subset selection. The two main reasons for this are the following.

Firstly, we have to consider the dimensionality of the design space. In dispersion problems, the locations are often points in two- or three-dimensional space. Subset selection on the other hand applies to datasets of any dimension. In practice, the dimension is often considerably larger than 3. This means that we have to determine whether an algorithm for the dispersion problem extends to higher dimensions, before we can use it in subset selection.

Secondly, many algorithms only give a solution in a reasonable amount of time for sets containing relatively few (a couple of dozen or a few hundred) points (Agca et al. 2000, Pisinger 2006). Subset selection on the other hand is generally applied when there are thousands of points. Ideas which give no computational problems for many dispersion problems could require too much time or memory capacity to be useful for subset selection problems.

Keeping these two aspects in mind, we can apply some methods for the dispersion problem directly to subset selection. The greedy MAXMIN method (Ravi et al. 1991) is one such method. It starts by selecting the two points furthest away from each other. Then iteratively, the point furthest away from the already selected points is added to the subset. Although this method is quite simple, Ravi et al. (1991) have show that if the triangle inequality holds, the heuristic gives an approximation ratio of 2. This means that we can guarantee that the MAXMIN distance of the subset we obtain is at most twice the MAXMIN distance of the optimal subset. Furthermore, due to its simplicity the method is also quite fast and thus suitable for large datasets. Therefore, we also use this method for uniform subset selection and describe it in more detail in Section 3.3.

1.4 Overview

The structure of the rest of this paper is as follows. In Section 2, we show with a simple example that taking a subset can indeed improve the aspects mentioned in this introduction. New and current methods to select a subset are described in Section 3. The main difference between the new and current methods is that the new method also uses the output data. To see if this results in subsets which produce beter Kriging models, we make a comparison between the methods in Section 4. Furthermore, we make a comparison with radial basis function (RBF) models fitted to the complete dataset. The advantage of RBF-models is that they have shown good fits to both stochastic and deterministic functions (Powell 1987) and that they require significantly less time to be fitted than Kriging models (Jin et al. 2001). RBF-models can therefore be fitted to datasets for which Kriging models would be too time-consuming. In Section 4, we thus also compare the performance, in terms of time-consumption and accuracy, of Kriging models fitted to a subset and RBF-models fitted to the complete dataset. A basic description of RBF-models can be found in Appendix F. Finally, Section 5 contains the conclusions and suggestions for further research.

2 Example

To show that selecting a subset can really improve the aspects mentioned in Section 1.1, we introduce the following simple artificial example. We try to approximate the six-hump camel-back function (Dixon and Szegö 1978):

$$f(x) = 4x_1^2 - 2.1x_2^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4,$$

with $x_1 \in [-2, 2]$ and $x_2 \in [-1, 1]$. As our dataset, we take a maximin LHD of 20 points (Van Dam et al. 2007) with four additional points close to an existing point as depicted in Figure 1. By adding these four points, we have created a cluster of points in the dataset. As mentioned in Section 1.1 this can cause several problems.

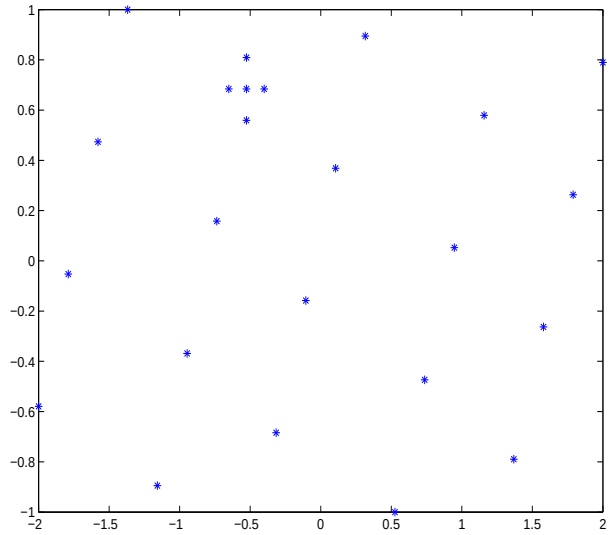

Figure 1: Maximin LHD of 20 points with 4 additional points.

We test the effect of selecting a uniform subset by fitting Kriging models to the datasets with and without the four additional points. To measure the effects of taking a subset on the different aspects, we use the performance measures which are described in Section 4.2. The results of the performance measures of both Kriging models are given in Table 1.

	Without four additional points	With four additional points
RMSE	0.48	0.51
Maximum Error	2.03	2.46
Condition Number	76	1491564
Average Robustness	0.97	8.36
Max. Robustness	1.57	100.66

Table 1: Performance of Kriging models fitted to the datasets with and without the four additional points.

Both the RMSE and the Maximum Error show that the accuracy of the Kriging model without the four points is better than that of the model with the additional four points. The latter Kriging model focusses more on fitting accurately in the region of these additional points and is, as a result, more accurate in this region. However, this additional accuracy is at the expense of accuracy in other regions, which deteriorates the overall accuracy.

When we compare the condition numbers, we see a very large difference. This shows that the additional 4 points make the resulting Kriging model much more susceptible to numerical inaccuracies. Finally, the larger maximal and average robustness values indicate that this model is also less robust with respect to errors in the output data.

This simple example thus shows that removing some points can improve the quality of the resulting Kriging model. Determining which points to remove is quite easy in this case. In practice when the number of points is much larger, this becomes less straightforward. Therefore, we introduce in the next section some current and new methods to select points from a dataset.

3 Subset selection methods

3.1 Orthogonal Array Selection

In the paper by Srivastava et al. (2004), the problem is discussed of selecting 500 or fewer points from a dataset containing 2490 points in 25-dimensional space. The selected points are used to create a Kriging model and the remaining points are used to check the accuracy of this model. Results are compared for different numbers of selected points and with results obtained by using quadratic response-surface models.

To select the points, first a randomized orthogonal array is constructed. Then for each point of the orthogonal array a "nearest neighbor" is determined, i.e. the data point closest to the orthogonal array point. All "nearest neighbors" together form the set of selected points. It is possible that this set contains less points than the orthogonal array because one point in the dataset can be the "nearest neighbor" of multiple points of the orthogonal array. As we do not know in advance how often this happens, we cannot set the subset size exactly. However, we do know that the size of the orthogonal array is an upper bound for the subset size.

The idea of selecting "nearest neighbors" to points of an orthogonal array can be extended to other types of space-filling DoCEs. We could for instance also use maximin LHDs. In order to limit the number of methods compared in this paper, we do not use Maximin LHD Selection as a separate method. We do however use it to generate starting solutions for the Sequential Selection methods described in Section 3.5. The (approximate) maximin LHDs we use for this are obtained with the ESE-algorithm of Jin et al. (2005).

Another reason for using Orthogonal Array Selection is that it enables us to compare our results with those found by Srivastava et al. (2004). It is however unclear which method they exactly used to construct the randomized orthogonal arrays. We have decided to use the method described on page 131 in Hedayat et al. (1999) to construct 61 and 113-dimensional orthogonal arrays of respectively 250 and 686 points. Using a uniform distribution, we randomly select 6 or 25 dimensions to obtain randomized orthogonal arrays that are suitable for our test problems. As different orthogonal arrays can result in different subsets, the choice of orthogonal array could affect the quality of the resulting subset. We therefore determine for each dataset ten subsets using ten different orthogonal arrays. We thus hope to determine whether the choice of orthogonal array indeed affects the quality of the subset and the resulting model.

Notice, that this method requires us to find a suitable orthogonal array. This could be a problem as an orthogonal array of the desired number of points and dimensions might not be known. However, due to the flexibility in the desired number of points of the orthogonal array, this problem can in most cases easily be resolved.

3.2 Fast Exchange Algorithm

The Fast Exchange Algorithm was introduced by Lam et al. (2002) to select a subset from a very large database containing characteristics of molecules. The aim of the algorithm is also to select a subset such that it covers the numerical space described by the database. However, as this space is often high-dimensional, nearly all of these databases are sparse. This makes it impossible to densely cover the whole space with a reasonably sized subset of points. The algorithm therefore focusses on selecting the subset such that it is space-filling in low-dimensional projections of the space.

The algorithm globally works as follows. First, sets of bins are constructed for all 1-D, 2-D and 3-D projections. These bins form a partition of the projected design spaces and their sizes depend partly on the distribution of the data. The aim is now to select a subset such that each bin contains approximately the same number of points. To measure how close a certain subset is to this aim, the uniform cell coverage criterion is used. This criterion has the advantage that we can easily calculate the effect of adding or removing a particular point from the subset on the criterion value.

The Fast Exchange Algorithm now tries to find the best subset by exchanging points that are inside and outside the subset. This is done in two steps. In the first step, the best point to add to the current subset is determined. The

resulting subset thus has a size of $n+1$. In the second step, the best point to remove from this subset is determined. The main difference with the basic exchange algorithm is in these two steps. In the basic exchange algorithm, all possible points outside the subset are tried in the first step and all points inside the subset in the second step. We thus loop through all points in the dataset before an exchange is made. The Fast Exchange Algorithm however uses a distribution of the improvements to select the exchange. At the begin of the algorithm, 100 additions and removals of points are tried to estimate the two distributions of the improvements in Steps 1 and 2. During the algorithm, when more additions and removals are tried, this estimated distribution is updated. A point is now added or removed if it belongs to the upper tail of this distribution. This often implies that we loop through a lot less points before an exchange is made.

3.3 Greedy MAXMIN Selection

The greedy MAXMIN method comes from the field of dispersion problems. In the description of the algorithm, k-dimensional data points are therefore seen as points in k-dimensional space. As the name indicates, it seeks to maximize the minimal Euclidean distance between any two points in the chosen subset. It does this in essentially the same way as the "furthest point outside the neighborhood" heuristic described in Steuer (1986).

Let us denote the total dataset by N and the Euclidean distance between points i and j by $d_{i,j}$. We can then describe this heuristic as follows: (Ravi et al. 1991)

1. Take $S = \emptyset$.
2. Let (i, j) be such that $d_{i,j}$ is maximal.
3. Add i and j to the set S.
4. Find a point $i \in N \backslash S$ such that $\min_{j \in S} d_{i,j}$ is maximum among the points in $N \backslash S$.
5. Add point i to S.
6. Repeats Step 4 and 5 until the set S contains the desired number of points.

In Step 4, we thus determine the point furthest away from the already chosen points. Although the heuristic is quite simple, Ravi et al. (1991) have shown that if the triangle inequality holds, the heuristic gives an approximation ratio of 2. Furthermore, they have proven that, unless P=NP, no polynomial-time relative approximation algorithm can provide a better performance.

Notice, that if the domains of the variables are of different magnitude, it is better to normalize the domains in order to obtain a uniform subset. For the artificial datasets used in this paper, this is not necessary as the domains for all variables are the same. This is however not the case for the variables in the HSCT dataset, so we will normalize them before applying this method.

3.4 Greedy DELETION Algorithm

Besides MAXMIN Selection, we also use another simple greedy algorithm. This greedy method constructs a subset by iteratively removing one point of the pair of points with the smallest Euclidean distance between them. To decide which of the two points should be removed, we look for each point at the distance to its second closest point. The point for which this distance is smallest is removed from the dataset.

Using the same notation as for MAXMIN Selection, we can describe the DELETION Algorithm as follows:

1. Take $S = N$.
2. Let (i, j) be such that $d_{i,j}$ is minimal.
3. Determine $c_i = min_{k \in S/\{i,j\}} d_{i,k}$ and $c_j = min_{k \in S/\{i,j\}} d_{j,k}$.
4. Remove the point with the lowest c value from the set S.
5. Repeats Steps 2, 3 and 4 until the set S contains the desired number of points.

As this method constructs a subset by removing points from the total dataset, the method requires less iterations for larger subsets than for smaller. This in contrast to for instance MAXMIN and Sequential Selection which build a subset by adding points to an initially empty set. Furthermore notice that also for this method it is useful to normalize the domains of the variables in order to obtain a uniform subset.

3.5 Sequential Selection

In the three methods discussed in the previous section, the output values of the points are not used in the selection process. However as the selected points are used to determine a model of the output, it seems a good idea to explicitly use the output information in selecting the training points. As previously mentioned, this is an important difference with DoCE where generating the output values is expensive. In our situation, we work with a given dataset with known output values and can thus use the output values at no additional computational costs.

We use the output values in the following way. First, we apply Maximin LHD Selection (see Section 3.1) using only the input values to determine an initial training set. Then we fit a Kriging model to the training set and calculate the prediction error at the non-selected points. The idea is now to add non-selected points with a large error to the training set. We should however take into account that if the Kriging model is inaccurate in a certain region, all points in that region have a large error. Thus simply selecting, for instance, n_1 points with the highest error might result in adding points that are clustered together in one or a couple of regions.

To reduce this problem, we use two methods. The first method is to add one point at a time. We thus add the point with the largest error to the training set and then refit the Kriging model. This method completely solves the problem, but is unfortunately quite time consuming as we fit a new Kriging model after every added point. This is also the reason that we developed a second method. This method determines the $n_2 > n_1$ worst points and then uses the greedy MAXMIN heuristic described in Section 3.3 to select a uniform subset of n_1 points. These n_1 points are then added to the training set. In this paper, we use $n_1 = 10$ and $n_2 = 40$ to test this method. We have not tested the influence of the choice of n_1 and n_2 on the quality of the resulting subset. Determining this possible influence and a suitable method for selecting these values thus remain topics for further research. To determine whether the above problem really occurs and results in worse models, we also test the method where we simply add the $n_1 = 10$ worst points.

For each method, the selection and addition step are repeated until the desired number of points are selected. Notice however that because we use Maximin LHD Selection to determine the initial training set, the initial training set might not always be equal to the size of the LHD. We take this into account when determining the number of selection and addition steps. As for two methods we add 10 points per step, it is not always possible to determine this such that we exactly get the desired number of points. In these cases, we select the largest attainable number of points smaller than the desired number of points.

As it is a sequential method, we can also decide to use another stopping criterion to determine how often we repeat the steps. An example would be to stop when the maximum error in the non-selected points is below a certain predefined value. In this paper, we choose to predefine the number of selected points for easier comparison between the different methods.

Finally, we want to make a remark on the implementation of this method. As mentioned, the method requires that a new Kriging model is fitted in every iteration. As this is the most time consuming part of the method, we tried to speed up the optimization of the Kriging parameters. We managed to do this by using the fact that subsets in consecutive iterations are quite similar as one is obtained by adding points to the other. This fact makes it quite plausible that the optimal parameter settings of the Kriging models fitted to both subsets are also quite similar. The optimal parameter setting of the Kriging model in the previous iteration therefore seems a good starting solution for fitting the Kriging model in the current iteration. Some tests show that using the optimal parameters of the previous iteration instead of a fixed starting solution, can indeed reduce the time needed to select a subset by 15 percent. Therefore, we use this implementation for the Sequential Selection method in the remainder of the paper.

4 Computational results

4.1 Tested subset selection methods

The methods that we will compare are the following:

- Random Selection (RS).
- Orthogonal Array Selection (OAS).
- Fast Exchange Algorithm (FEX).
- Greedy MAXMIN Selection (MAXMIN).
- Greedy DELETION Algorithm (DELETION).
- Sequential Selection adding one point at a time (SS1).
- Sequential Selection adding ten worst points at a time (SS10).

- Sequential Selection adding ten uniform points from the 40 worst points at a time (SS1040).

The first method randomly selects the required number of points without taking into account any of their properties. The performance of this method is used as a reference for the performance of the other methods.

To also test the effect of the training set size on the resulting Kriging model, we select subsets of 250, 350 and 500 points from the datasets described in Section 4.3. As the SS1 method is quite time consuming, we will only generate subsets of 250 and 350 points using this method. Furthermore for Orthogonal Array Selection, we cannot determine the exact subset size in advance. By using orthogonal arrays of 250 and 686 points, we get subsets of at most these amounts of points. As mentioned in Section 3.1, we generate subsets using ten different orthogonal arrays to test the effect of the choice of orthogonal array on the resulting subset. For each performance measure, we therefore report the mean, minimum, and maximum over these ten subsets.

4.2 Performance measures

The different motivations for selecting a subset require that we use several performance measures to determine how good a certain subset is. We describe the used performance measures and our motivation for choosing them.

RMSE and Maximum Error

Our first motivation is the selection of a subset that results in an accurate model. We thus need to measure the accuracy of the resulting Kriging model. Common measures are the Average Error, Root Mean Squared Error (RMSE), and Maximum Error. We cannot measure these on the training data because the Kriging model will interpolate through the training data. Instead, we need a validation set which could either be the remaining dataset or a separately generated set.

In the case of a real-life dataset, we only have the first option. If the original dataset is non-uniform, the remaining dataset might not be particularly suited to measure the overall accuracy of the model as the accuracy in densely populated regions weighs heavier than accuracy in sparsely populated regions. This problem could be reduced by taking the Weighted Average Error or Weighted RMSE. The weights should then be determined such that errors in points in sparse regions get more weight than errors in points in dense regions. It is however unclear how exactly the weights should be determined to achieve this effect. Therefore, we will use the usual RMSE in this paper, although we are aware of its deficiency. Finding a method for determining the weights remains a worthwhile subject for further research. Besides the RMSE, we will also use the Maximum Error to compare the accuracy of different Kriging models.

For artificial datasets, the best option is to separately generate a uniform validation set. This way we avoid any problems with densely and sparsely populated regions. In this paper, a uniform grid is therefore used as the validation set.

Time model fitting

To determine the reduction in time necessary to fit the Kriging model, we simply use the amount of required CPU-time. For fitting the Kriging model, we used the Matlab toolbox DACE provided by Lophaven et al. (2002). All of the calculations where performed on a PC with a 2.4-GHz Pentium 4 processor. The results in this paper are all reported in minutes.

Time subset selection

Besides fitting the Kriging model, also the time necessary for constructing the subset is measured. This is needed to determine if the time gained for model fitting is not outweighed by the additional time needed to select a good subset. Note that all subset selection methods are coded by the author in Matlab. By using the same program for each method, we aim to make a fair comparison between the different methods. The results are again reported in minutes.

Condition number

We also want to avoid ill-conditioned correlation matrices by selecting a subset. To determine if a correlation matrix is ill-conditioned, we can use its condition number. The condition number κ of a square matrix C is defined as (Golub and Van Loan 1996):

$$\kappa(C) = \frac{\sigma_{\max}(C)}{\sigma_{\min}(C)},$$

where $\sigma_{\max}(C)$ and $\sigma_{\min}(C)$ are the largest and smallest singular values of C. The condition number is a measure of the worst case loss in precision when solving a linear system. A matrix with a large condition number is called ill-conditioned and is thus susceptible to numerical inaccuracies.

Average and maximum robustness

Finally, we also want to measure the influence on robustness with respect to errors in the output data. For Kriging models, Siem and Den Hertog (2007) suggest to use $||c(x)||$ as a measure of robustness in the point x, where $c(x)$ is the vector of Kriging weights (See Appendix E for a basic description of the Kriging method). This measure gives an indication of the factor by which an error in the data point x may be enlarged by the Kriging model. A higher value of this criterion thus means a lower robustness.

This robustness-criterion only determines the robustness at a certain point x. To measure the overall robustness, we use the maximal and average value of the robustness values in a set of points. As we do not need to know the real output values in these points, we will select them on a uniform grid even in the case of a real-life dataset.

4.3 Datasets

To test the different selection methods, we use two types of datasets: artificial datasets with known underlying function and real-life datasets for which the underlying function is unknown. For real-life datasets, it is clearly more difficult to judge the accuracy of the resulting model, especially as these datasets are often unstructured and non-uniform. This also holds for the real-life dataset used in this paper. The dataset was originally used in the design of the High Speed Civil Transport (HSCT) aircraft and contains 2490 points in 25-dimensional space. As this dataset is also used by Srivastava et al. (2004), we use it to make a comparison with their results. Before using it, we however first removed the duplicate points which left us with a dataset of 2487 unique points.

Artificial datasets are generally constructed by drawing points from the design space and calculating their value for a known function. In this paper, the six-variable Hartman-6 function (Dixon and Szegö 1978) is used to generate the output data. This function is defined as follows:

$$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} \alpha_{ij}(x_j - p_{ij})^2\right)$$

with $x_j \in [0,1]$, $j = 1,\ldots,6$, and where the parameters are given in the following table:

i	α_{ij}, $j=1,\ldots,6$						c_i	p_{ij}, $j=1,\ldots,6$					
1	10	3	17	3.5	1.7	8	1	.1312	.1696	.5569	.0124	.8283	.5886
2	.05	10	17	0.7	8	14	1.2	.2329	.4135	.8307	.3736	.1004	.9991
3	3	3.5	1.7	10	17	8	3	.2348	.1451	.3522	.2883	.3047	.6650
4	17	8	.05	10	0.1	14	3.2	.4047	.8828	.8732	.5743	.1091	.0381

The output values of the Hartman-6 function range between approximately -3.23 and 0. The Hartman-6 function is chosen because it is a widely used test problem with a relatively large number of dimensions (Wang et al. 2001) (Jin et al. 2002). As most real-life datasets are also high dimensional, we prefer this test problem over other widely used but lower dimensional test problems.

To draw the points from the design space, the following methods are used:

1. Draw the values of all coefficients from a uniform distribution on the range of each variable.
2. Divide the design space into m^d equally sized cells, where d is the dimension. Determine m^d numbers that sum up to the total size of the dataset. Randomly assign these numbers to the cells. For each cell use a uniform distribution to sample the assigned number of points within the cell. By adjusting the distribution of the number of points, we can vary the uniformity of the dataset.

We refer to datasets constructed with the first method as uniform datasets. We use the second method to construct non-uniform datasets of 2000, 5000, and 10000 points. For all sizes, m is set equal to 3 which means that we have 729 cells. The distributions of the number of points per cell are given in Table 2.

For 2000 points, it is still possible to fit a Kriging model and a RBF-model to the complete dataset. We thus use these datasets to test if taking a subset is really better than using the complete set. For the datasets of 5000 points, we were no longer able to fit a Kriging model to the complete dataset with the DACE toolbox. A RBF-model could however still be fitted to the complete dataset. Therefore, we use these datasets to test how a Kriging model fitted to a subset compares to a RBF-model fitted to the complete set. When using 10000 points, we could no longer fit a RBF-model to the complete dataset. We thus use these datasets to compare the results of the different subset selection methods for Kriging only.

Number of cells	30	30	74	238	431	595
Points per cell 2000 points	20	10		2	1	
Points per cell 5000 points	50	25	5			4
Points per cell 10000 points	100	50	10			8

Table 2: Distributions of the number of points per cell used to construct non-uniform datasets of 2000, 5000, and 10000 points.

For each size and type of artificial dataset, we randomly generated 50 datasets. In the comparisons, we use the average, minimum, and maximum performance over these datasets for all performance measures.

4.4 Results for artificial datasets of 2000 points

For datasets of 2000 points, it is still possible to fit a Kriging and a RBF-model to the complete dataset. By taking a subset, we however aim to improve some of the performance measures. In Table 3 in Appendix A, we have an overview of the results of the different performance measures.

When looking at the RMSE and the Maximum Error, the Kriging model fitted to the complete dataset is the most accurate model for both uniform and non-uniform datasets. Unfortunately, it also requires by far the most time to fit. Fitting a RBF-model is much faster than Kriging and also faster than most subset selection methods if we take into account the times necessary to select the subsets. However, some of the Kriging models fitted to a subset are more accurate than the RBF-model. For subsets of 200 points, the RMSE of most Kriging models is still worse, but SS1, SS10 and SS1040 already produce almost equally good of even slightly better Kriging models. For 350 points SS1, SS10 and SS1040 result in Kriging models with a really lower RMSE than the RBF-model. When using 500 points, these methods result in Kriging models with even lower RMSE values, as expected. For non uniform datasets, also the best subsets obtained with OAS result in Kriging models with a lower RMSE. The Maximum Error shows even more cases where Kriging models fitted to subsets are better than the RBF-model.

When we consider the subset selection times, two methods that require hardly any time are RAND and OAS. Using RAND is however not a very good choice, as the resulting Kriging models are quite inaccurate. The OAS method results in considerably more accurate Kriging models. Subset methods which result in even more accurate Kriging models are SS1, SS10 and SS1040. When we compare these methods for each subset size separately, SS1 results in the lowest RMSE and SS1040 in the lowest Maximum Error. When we however also take into account the time to select the subset, SS1 is the least favorable of these methods. Selecting a subset of 350 points even requires more time than fitting a Kriging model to the complete dataset. The methods SS10 and SS1040 are also relatively time-consuming compared to RAND, MAXMIN, FEX and OAS, but they are more accurate and have still a considerable time reduction compared to Kriging fitted to the complete dataset.

Considering the condition numbers, SS1, SS1040, MAXMIN, DELETION and OAS perform very well. The good performance of the latter two methods might be explained by the fact that they only focus on optimizing the uniformity of the input data which influences the condition number of the correlation matrix. It is however remarkable that the FEX method, which has the same focus, performs considerably worse. This seems to indicate that the FEX does not succeed as well in selecting a uniform subset as SS1, SS1040, MAXMIN, DELETION and OAS do. Furthermore, it is nice to see that SS1040 gives better result than SS10 as expected. Taking 10 uniformly selected points from the 40 worst instead of just taking the 10 worst, thus seems to result in a more uniform subset. Most important however is that all subset selection methods result in a correlation matrix with a considerably lower condition number than that of the correlation matrix of the complete dataset. Taking a dataset thus clearly reduces the chance of numerical inaccuracies.

For robustness, the same conclusion can be drawn that taking a subset considerably improves the performance measure. When we look at the maximum robustness, SS1, SS1040, MAXIMIN, DELETION and OAS perform best. For the average robustness also SS10 performs well.

Besides these performance measures, we also present the real sizes of the subsets. We do this because for OAS the number of points in the subset is not known exactly in advance. The results show that for 250 points, the number of selected points is quite close to the size of the orthogonal array. For the orthogonal arrays of 686 points, the difference is much larger. As mentioned in Section 3.5, SS10 and SS1040 can also result in slightly different subset sizes as Maximin LHD Selection is used for the initial training set. The results however show that this does not occur for the tested datasets.

When we make a comparison between uniform and non-uniform datasets, we see almost the same results for most performance measures. The ranking of the different subset selection methods is also generally the same. The main

differences are between the condition number and robustness of the Kriging models fitted on the complete dataset. On these aspects, the Kriging models fitted on the uniform datasets performs better than the Kriging models fitted on the non-uniform datasets. For the Kriging models fitted on the subsets, the differences on these aspects are much smaller. This seems to indicate that the difference in uniformity between the subsets is smaller than the difference in uniformity between the two types of complete sets.

A factor which also influences the performance measures is the subset size. When we compare the performance measures, we see two opposite effects. The RMSE and Maximum Error improve when larger subsets are used and the other performance measures worsen. To determine the 'best' subset size, we thus have to make a trade-off between accuracy and robustness, required time and numerical accuracy. Which trade-off is best depends on the importance of these aspects and the estimated accuracy of the dataset.

Finally, we take a look at the effect of the choice of orthogonal array on the results of the OAS method. We do this by comparing the mean, minimum, and maximum of the performance measures over ten subsets obtained with ten different orthogonal arrays. The differences in these results show that the choice of orthogonal array has a clear effect on the quality of the resulting subset and Kriging model. When using the OAS method, it is thus better not to use just one random orthogonal array. It would be better to carefully choose a suitable orthogonal array or to create multiple subsets using different (random) orthogonal arrays and then to select the best subset. Which option works best and how exactly we should implement them is not immediately clear and is thus an interesting subject for further research.

4.5 Results for artificial datasets of 5000 and 10000 points

For datasets of 5000 points, we could no longer fit a Kriging model to the complete dataset. A RBF-model could however still be fitted. As most results are quite similar to the results of the datasets containing 2000 points, we mainly discuss the differences. Table 4 containing the results can be found in Appendix B.

When comparing the RMSE, we see that now only SS10 and SS1040 with 500 points result in Kriging models with approximately equal RMSE values as the RBF-model. The Maximum Error is however considerably lower for all subsets of SS1, SS10 and SS1040 and for the large subsets obtained with OAS. Combining these two measures, the Kriging models based on subsets of 500 points obtained with SS10 and SS1040 seem to be the most accurate. The time required to select these subsets and to fit the model is unfortunately considerably more than for the RBF-model. We thus have a trade-off between accuracy and the required time. For the condition number and robustness, we see the same result as for the datasets of 2000 points. Considering the subset sizes, the only difference is that the subsets obtained with OAS have become larger.

The above results also apply to the datasets of 10000 points. The only difference is that for 10000 points, it was not possible for us to fit a RBF-model. A comparison with a model fitted to the complete dataset could thus no longer be made. All the results can be found in Table 5 in Appendix C.

4.6 Results for HSCT dataset of 2487 points

To make a comparison with the results of Srivastava et al. (2004) on the HSCT dataset, we have to use some different performance measures. In Srivastava et al. (2004), the Root Mean Squared Percentage Error (RMSPE), the Average Percentage Error and the Maximum Percentage Error are used to measure the accuracy. We therefore report these accuracy measures for the different subsets instead of the Maximum Error and the RMSE .

Another difference is in the subset sizes as Srivastava et al. (2004) find subsets of 126, 283 and 372 points. To make a fair comparison, we compare our subsets of 250 points with the subset of 283 points and the subsets of 350 points with the subset of 372 points. The subsets of 500 points are thus only used for the comparison between the different methods in this paper.

When comparing the results of Srivastava with our results, one of the surprising results is the difference in subset sizes when using the OAS method. Srivastava reports that using random orthogonal arrays of 250 and 686 points results in subsets of 126 and 283 points respectively. When we however use orthogonal arrays of these sizes, we obtain subsets of approximately 225 and 500 points. We tried to determine the cause of this large difference, but were not able to find a satisfying explanation. Use of different orthogonal arrays could be one explanation, but it seems unlikely that this alone causes such a large difference.

Looking at the other performance measures for the subsets of 250 and 283 points, we see that the RMSPE and the Average Percentage Error obtained by Srivastava are equal to the results obtained with RAND. This is surprising as RAND is a very naive method. Furthermore, SS1, SS10 and SS1040 perform better on these two performance measures. For the Maximum Percentage Error, besides these methods also MAXMIN and OAS give better results. Although OAS is the same method as applied by Srivastava, these results are difficult to compare because of the difference in the resulting subset sizes.

For the subsets of 350 an 372 points, Srivastava obtained the best results for the RMSPE and Average Percentage Error. Of the other methods, SS1 and SS1040 also perform well on these measures. The Maximum Percentage Error

is lowest for the SS1 method, followed by Srivastava's results and SS1040. Notice, that SS10 performs quite bad on the Maximum Percentage Error and in a lesser extend on the RMSPE. This could be a sign that indeed a cluster of points is selected which negatively affects the overall accuracy.

As the other performance measures are not calculated by Srivastava, we can only compare them over the different methods we implemented ourselves. For the condition number and average robustness, SS1, SS1040, MAXMIN, DELETION and OAS give good results. The high values for the condition number and the average robustness measure for the SS10 method are again a sign of possible clustering.

The subset selection times are again lowest for RAND and OAS, followed by MAXMIN and DELETION. The time needed to fit the model is lowest for SS1, SS10 and SS1040. This is the result of using the optimal parameters settings of previous iterations of the algorithm. If we would not have used this information, but instead used a standard starting vector, the time to fit the model would be comparable to those of the RAND method. All in all, the results are similar to our results for the three artificial datasets. Although we used some different performance measures for this datasets, the ranking of the different methods for most aspects is the same. This increases our confidence that results obtained by the artificial datsets can also be obtained for real life datasets.

5 Conclusions and further research

In this paper, we show that fitting a Kriging model to a smaller but more uniform dataset can result in better Kriging models. Especially for large non-uniform datasets using a uniform subset instead of the complete dataset can have several advantages. Reducing the time necessary to fit the model, avoiding numerical inaccuracies and improving the robustness with respect to errors in the output data are some of the aspects which can be improved by using a more uniform subset.

To select a uniform subset several new and current methods are described. All these methods are tested on artificial subsets of three different sizes and two levels of uniformity. Furthermore, tests are performed on the HSCT dataset, which was also used by Srivastava et al. (2004). The tests show that by using uniform subsets, we can indeed find accurate Kriging models in less time. Furthermore, these Kriging models are more robust and less susceptible to numerical inaccuracies. When comparing the different methods for finding subsets, there is no overall winner. SS1040 generally performs well on accuracy, robustness and numerical accuracy. Subsets obtained with SS1040 even results in Kriging models which are more accurate than RBF-models fitted on the complete dataset. Compared to the other methods, SS1040 is however relatively time consuming, but still considerably faster than fitting a Kriging model to the complete dataset. The OAS method is much faster, but has a lower accuracy, robustness and numerical accuracy. Deciding which method is best for a practical application, thus depends on how the different aspects are valued. The comparison made in this paper can be used to facilitate the user in making a good choice.

Further research on this problem could be targeted at finding new methods for selecting a subset. These new methods could for instance try to find even better subsets by using a different objective to obtain a uniform subset or by taking into account other properties of the dataset. Another option would be to develop methods which dynamically determine a suitable subset size. Furthermore, it would be interesting to determine whether the results found in this paper also apply to other modeling methods. There could be more modeling methods which benefit from using a small uniform dataset instead of a large non-uniform one.

6 Acknowledgements

The author wishes to thank Edwin van Dam and Dick den Hertog for the many inspiring conversations on the topic of this paper, and Peter Stehouwer and Erwin Stinstra for some useful ideas. Furthermore, the author wishes to thank Kemper Lewis, Christina L. Bloebaum and Timothy W. Simpson for providing the HSCT dataset.

References

Agca, S., B. Eksioglu, and J.B. Ghosh (2000). Lagrangian solution of maximum dispersion problems. *Naval Research Logistics*, **47(2)**, 97–114.

Banzhaf, W., F.D. Francone, R.E. Keller, and P. Nordin (1998). *Genetic Programming: An introduction on the automatic evolution of computer programs and its applications.* Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Booker, A.J., J.E. Dennis, P.D. Frank, D.B. Serafini, V. Torczon, and M.W. Trosset (1999). A rigorous framework for optimization of expensive functions by surrogates. *Structural and Multidisciplinary Optimization*, **17(1)**, 1–13.

Cherkassky, V. and F. Mulier (1998). *Learning from data : Concepts, theory, and methods.* New York, NY, USA: John Wiley & Sons, Inc.

Dam, E.R. van, B.G.M. Husslage, D. den Hertog, and J.B.M. Melissen (2007). Maximin Latin hypercube designs in two dimensions. *Operations Research*, **55(1)**, 158–169.

Davis, G.J. and M.D. Morris (1997). Six factors which affect the condition number of matrices associated with Kriging. *Mathematical Geology*, **29**, 669–683.

Dixon, L.C.W. and G.P. Szegö (1978). The global optimization problem: An introduction. In L.C.W. Dixon and G.P. Szegö (Eds.), *Toward Global Optimization*, Volume 2, 1–15. North-Holland.

Erkut, E. (1990). The discrete p-dispersion problem. *European Journal of Operational Research*, **46**, 48–60.

Erkut, E. and S. Neuman (1989). Analytical models for locating undesirable facilities. *European Journal of Operational Research*, **50**, 275–291.

Ghosh, J.B. (1996). Computational aspects of the maximum diversity problem. *Operations Research Letters*, **19**, 175–181.

Golbraikh, A., M. Shen, Z. Xiao, Y.D. Xiao, K.H. Lee, and A. Tropsha (2003). Rational selection of training and test sets for the development of validated QSAR models. *Journal of Computer-Aided Molecular Design*, **17(2)**, 241–253.

Golbraikh, A. and A. Tropsha (2002). Predictive QSAR modeling based on diversity sampling of experimental datasets for the training and test set selection. *Journal of Computer-Aided Molecular Design*, **16(5-6)**, 357–369.

Golub, G.H. and C.F. van Loan (1996). *Matrix computations (3rd edition)*. Baltimore: Johns Hopkins University Press.

Hansen, P. and I.J. Moon (1994). Dispersing facilities on a network. *Cahiers du CERO*, **36**, 221–234.

Hardy, R.L. (1971). Multiquadratic equations of topography and other irregular surfaces. *Journal of Geophysical Research*, **76(8)**, 1905–1915.

Hedayat, A.S., N.A.J. Sloane, and J. Stufken (1999). *Orthogonal arrays: Theory and applications*. New York: Springer.

Husslage, B.G.M., E.R. van Dam, D. den Hertog, H.P. Stehouwer, and E. Stinstra (2003). Coordination of coupled black box simulations in the construction of metamodels. *Concurrent Engineering*, **11(4)**, 267–278.

Jin, R., W. Chen, and T. W. Simpson (2001). Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, **23**, 1–13.

Jin, R., W. Chen, and A. Sudjianto (2002). On sequential sampling for global metamodeling in engineering design. *DETC-DAC34092, 2002 ASME Design Automation Conference*, 1–10.

Jin, R., W. Chen, and A. Sudjianto (2005). An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, **134(1)**, 268–287.

Jones, D.R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, **21(4)**, 345–383.

Jones, D.R., M. Schonlau, and W.J. Welch (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, **13(4)**, 455–492.

Koehler, J.R. and A.B. Owen (1996). Computer experiments. *Handbook of Statistics*, **13**, 261–308.

Kordon, A. (2006). Evolutionary computation at Dow Chemical. *SIGEVOlution*, **1(3)**, 4–9.

Koza, J.R. (1992). *Genetic Programming: On the programming of computers by natural selection*. MIT Press, Cambridge, MA, USA.

Krige, D.G. (1951). A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, **52(6)**, 119–139.

Kuo, C.C., F. Glover, and K.S. Dhir (1993). Analyzing and modeling the maximum diversity problem by zero-one programming. *Decision Sciences*, **24**, 1171–1185.

Lam, R.L.H., W.J. Welch, and S.S. Young (2002). Uniform coverage designs for molecule selection. *Technometrics*, **44**, 99–109.

Lophaven, S.N., H.B. Nielsen, and J. Sondergaard (2002). DACE: A Matlab Kriging toolbox version 2.0. Technical Report IMM-TR-2002-12, Technical Univeristy of Denmark, Copenhagen.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, **58(8)**, 1246–1266.

Pisinger, D. (2006). Upper bounds and exact algorithms for p-dispersion problems. *Computers and Operations Research*, **33(5)**, 1380–1398.

Powell, M.J.D. (1987). Radial basis functions for multivariable interpolation: A review. *Clarendon Press Institute of Mathematics and its Applications Conference Series*, 143–167.

Ravi, S.S., D.J. Rosenkrantz, and G.K. Tayi (1994). Heuristic and special case algorithms for dispersion problems. *Operations Research*, **42**, 299–310.

Ravi, S. S., Daniel J. Rosenkrantz, and Giri Kumar Tayi (1991). Facility dispersion problems: Heuristics and special cases (extended abstract). In *Algorithms and Data Structures, 2nd Workshop WADS '91, Ottawa, Canada, August 14-16*, 355–366.

Sacks, J., W.J. Welch, T.J. Mitchell, and H.P. Wynn (1989). Design and analysis of computer experiments. *Statistical Science*, **4**, 409–435.

Santner, Th.J., B.J. Williams, and W.I. Notz (2003). *The design and analysis of computer experiments*. Springer Series in Statistics. New York: Springer-Verlag.

Siem, A.Y.D. and D. den Hertog (2007). Kriging models that are robust with respect to simulation errors. *CentER Discussion Paper 2007-68*. Tilburg University.

Simpson, T. W., J. Peplinski, P.N. Koch, and J.K. Allen (2001). Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, **17**, 129–150.

Srivastava, A., K. Hacker, K. Lewis, and T.W. Simpson (2004). A method for using legacy data for metamodel-based design of large-scale systems. *Structural and Multidisciplinary Optimization*, **28**, 145–155.

Stehouwer, H.P. and D. den Hertog (1999). Simulation-based design optimization: Methodology and applications. In *Proceedings of the First ASMO UK / ISSMO Conference on Engineering Design Optimization*, Ilkley, UK.

Steuer, R.E. (1986). *Multiple criteria optimization: Theory and application*. New York: John Wiley.

Wang, G., Z. Dong, and P. Aitchison (2001). Adaptive response surface method - A global optimization scheme for approximation-based design problems. *Journal of Engineering Optimization*, **33**, 707–734.

A Results for artificial datasets of 2000 points

	Uniform Datasets									Non-Uniform Datasets								
	Average over datasets			Maximum over datasets			Minimum over datasets			Average over datasets			Maximum over datasets			Minimum over datasets		
	RMSE			RMSE			RMSE			RMSE			RMSE			RMSE		
Kriging	0.06			0.08			0.06			0.07			0.08			0.06		
RBF	0.10			0.11			0.10			0.11			0.12			0.10		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	0.17	0.15	0.13	0.21	0.18	0.16	0.14	0.13	0.11	0.18	0.15	0.13	0.22	0.18	0.17	0.15	0.13	0.11
SS1	0.10	0.09		0.12	0.10		0.09	0.08		0.11	0.09		0.13	0.11		0.09	0.08	
SS10	0.11	0.09	0.08	0.13	0.11	0.09	0.10	0.08	0.07	0.11	0.09	0.08	0.14	0.11	0.09	0.10	0.08	0.07
SS1040	0.11	0.09	0.08	0.13	0.11	0.09	0.09	0.08	0.07	0.11	0.09	0.08	0.12	0.11	0.09	0.10	0.08	0.07
MAXMIN	0.16	0.14	0.11	0.19	0.19	0.14	0.14	0.12	0.10	0.17	0.14	0.12	0.20	0.17	0.14	0.14	0.12	0.10
FEX	0.17	0.15	0.12	0.21	0.19	0.15	0.14	0.13	0.11	0.17	0.15	0.13	0.20	0.17	0.14	0.15	0.13	0.11
OAS mean	0.17		0.11	0.19		0.12	0.17		0.11	0.17		0.11	0.18		0.12	0.16		0.11
OAS min	0.15		0.10	0.17		0.11	0.14		0.10	0.15		0.10	0.17		0.11	0.14		0.09
OAS max	0.20		0.13	0.24		0.14	0.18		0.11	0.19		0.13	0.21		0.14	0.17		0.12
	Maximum Error			Maximum Error			Maximum Error			Maximum Error			Maximum Error			Maximum Error		
Kriging	0.76			1.05			0.50			0.85			1.29			0.54		
RBF	1.31			1.58			0.96			1.36			1.68			1.09		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	1.62	1.41	1.28	2.05	1.98	1.74	1.15	0.85	0.86	1.65	1.54	1.42	2.02	1.96	1.81	1.12	0.99	0.84
SS1	1.00	0.95		1.52	1.43		0.60	0.51		1.10	1.05		1.56	1.53		0.73	0.70	
SS10	1.00	0.93	0.90	1.43	1.39	1.34	0.61	0.54	0.49	1.07	1.01	0.97	1.56	1.44	1.40	0.69	0.60	0.67
SS1040	1.01	0.95	0.92	1.46	1.44	1.37	0.59	0.53	0.60	1.05	1.02	0.98	1.51	1.51	1.47	0.68	0.63	0.65
MAXMIN	1.54	1.40	1.20	1.95	1.71	1.64	1.18	0.92	0.92	1.53	1.37	1.21	2.27	1.82	1.65	1.13	0.99	0.81
FEX	1.56	1.44	1.28	2.05	1.87	1.74	1.12	0.88	0.85	1.57	1.46	1.34	1.99	1.85	1.82	1.03	0.99	0.86
OAS mean	1.60		1.18	1.72		1.40	1.39		1.01	1.59		1.25	1.74		1.47	1.47		1.02
OAS min	1.28		0.93	1.51		1.23	0.98		0.71	1.30		1.00	1.59		1.30	0.99		0.78
OAS max	1.92		1.45	2.22		1.72	1.67		1.20	1.88		1.50	2.09		1.85	1.75		1.29
	Time Subset Selection			Time Subset Selection			Time Subset Selection			Time Subset Selection			Time Subset Selection			Time Subset Selection		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SS1	10.65	28.20		10.88	28.87		10.39	27.52		10.66	28.18		10.97	28.87		10.45	27.05	
SS10	1.03	2.75	8.14	1.07	2.80	8.34	1.00	2.67	7.97	1.03	2.74	8.13	1.05	2.82	8.35	1.02	2.66	7.92
SS1040	1.03	2.76	8.12	1.07	2.90	8.28	1.01	2.69	7.91	1.03	2.76	8.12	1.04	2.85	8.44	1.01	2.69	7.93
MAXMIN	0.43	0.59	0.83	0.43	0.60	0.84	0.42	0.58	0.82	0.43	0.59	0.83	0.43	0.60	0.84	0.42	0.58	0.82
FEX	0.21	0.28	0.33	0.37	0.47	0.70	0.08	0.09	0.11	0.24	0.30	0.39	0.35	0.42	0.57	0.15	0.17	0.20
OAS mean	0.00		0.01	0.00		0.01	0.00		0.01	0.00		0.01	0.00		0.01	0.00		0.01
OAS min	0.00		0.01	0.00		0.01	0.00		0.01	0.00		0.01	0.00		0.01	0.00		0.01
OAS max	0.00		0.01	0.01		0.01	0.00		0.01	0.00		0.01	0.00		0.01	0.00		0.01
	Time Model Fitting			Time Model Fitting			Time Model Fitting			Time Model Fitting			Time Model Fitting			Time Model Fitting		
Kriging	15.09			15.81			14.87			15.02			15.40			14.87		
RBF	0.25			0.26			0.25			0.25			0.26			0.25		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	0.10	0.21	0.51	0.11	0.23	0.53	0.09	0.20	0.48	0.10	0.21	0.51	0.11	0.23	0.53	0.10	0.20	0.48
SS1	0.10	0.21		0.11	0.22		0.09	0.20		0.10	0.21		0.10	0.22		0.10	0.20	
SS10	0.10	0.21	0.49	0.11	0.23	0.52	0.09	0.20	0.47	0.10	0.21	0.49	0.11	0.22	0.51	0.09	0.19	0.47
SS1040	0.10	0.21	0.49	0.10	0.22	0.51	0.09	0.20	0.47	0.10	0.21	0.49	0.10	0.22	0.55	0.09	0.20	0.45
MAXMIN	0.10	0.21	0.50	0.10	0.22	0.51	0.09	0.20	0.46	0.10	0.21	0.50	0.10	0.22	0.53	0.09	0.20	0.47
FEX	0.10	0.21	0.51	0.11	0.25	0.53	0.10	0.21	0.48	0.10	0.21	0.51	0.11	0.23	0.54	0.09	0.20	0.48
OAS mean	0.08		0.62	0.08		0.70	0.08		0.59	0.08		0.59	0.08		0.62	0.08		0.57
OAS min	0.07		0.50	0.08		0.60	0.06		0.47	0.07		0.47	0.07		0.53	0.06		0.42
OAS max	0.08		0.71	0.09		0.85	0.08		0.66	0.08		0.68	0.10		0.72	0.08		0.64

Table 3: Average, minimum, and maximum of all performance measures over 50 artificial datasets of 2000 points.

	Uniform Datasets						Non-Uniform Datasets					
	Average over datasets			Maximum over datasets			Minimum over datasets			Average over datasets			Maximum over datasets			Minimum over datasets
	Condition number			Condition number			Condition number			Condition number			Condition number			Condition number
Kriging	237859			935201			66964			420661			1409590			95175
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	1826	5367	8592	19835	47339	47927	355	495	651	1986	4194	11183	9879	17801	38230	314	1110	2384
SS1	246	538		551	1202		121	201		279	715		794	1530		127	264	
SS10	543	1109	2187	3048	7879	10806	166	331	697	670	1210	2766	3010	4043	6476	154	388	481
SS1040	222	453	1028	500	1089	2295	83	207	338	272	566	1630	602	1105	4046	83	222	509
MAXMIN	150	355	904	731	2062	3750	31	102	188	145	332	925	475	916	3062	28	100	237
FEX	919	2267	4898	2390	12665	14522	280	242	1199	922	2634	5159	3717	8739	27249	218	448	892
OAS mean	473		3991	1090		6902	273		2524	505		4222	958		9199	228		2072
OAS min	148		1351	271		2531	74		514	172		1419	299		2526	79		586
OAS max	1232		9668	4787		28866	434		4320	1275		9628	3344		28256	342		3972
	Maximum Robustness			**Maximum Robustness**			**Maximum Robustness**			**Maximum Robustness**			**Maximum Robustness**			**Maximum Robustness**		
Kriging	45.73			107.39			22.90			73.79			259.69			35.74		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	2.68	3.16	3.50	4.61	5.71	7.51	1.95	2.02	2.07	3.14	3.51	4.33	4.96	5.49	6.56	1.99	2.13	2.76
SS1	1.69	1.90		2.18	2.57		1.37	1.43		1.74	2.09		2.56	3.24		1.30	1.61	
SS10	2.31	2.54	2.73	4.61	6.09	5.56	1.52	1.66	1.81	2.49	2.67	2.99	4.21	4.35	5.05	1.63	1.84	2.13
SS1040	1.68	1.89	2.20	2.17	2.72	2.98	1.37	1.48	1.79	1.82	2.03	2.41	2.56	3.07	3.49	1.36	1.59	1.67
MAXMIN	1.35	1.45	1.64	1.78	1.89	2.22	1.10	1.21	1.30	1.35	1.44	1.67	1.66	1.83	2.24	1.10	1.21	1.30
FEX	2.33	2.55	2.95	3.73	3.75	4.09	1.64	1.81	2.14	2.33	2.87	3.07	3.85	6.05	4.97	1.71	2.08	1.97
OAS mean	2.18		3.13	2.59		3.68	1.92		2.64	2.25		3.26	2.85		3.98	1.91		2.71
OAS min	1.64		2.33	2.02		2.87	1.41		1.82	1.68		2.40	2.06		2.91	1.43		1.97
OAS max	2.96		4.37	4.01		7.11	2.23		2.98	3.21		4.74	5.38		8.11	2.25		2.99
	Average Robustness			**Average Robustness**			**Average Robustness**			**Average Robustness**			**Average Robustness**			**Average Robustness**		
Kriging	6.33			10.62			4.52			8.48			11.92			6.08		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	1.08	1.25	1.39	1.55	2.23	1.99	0.80	0.87	0.90	1.11	1.25	1.47	1.57	1.85	2.08	0.78	0.92	1.10
SS1	0.67	0.77		0.77	0.89		0.60	0.64		0.68	0.78		0.77	0.90		0.59	0.65	
SS10	0.73	0.83	0.96	0.81	0.96	1.11	0.63	0.70	0.78	0.75	0.84	0.99	0.92	0.96	1.16	0.63	0.71	0.78
SS1040	0.70	0.78	0.89	0.79	0.94	1.08	0.55	0.69	0.75	0.73	0.80	0.93	0.85	0.93	1.14	0.58	0.67	0.77
MAXMIN	0.82	0.89	1.01	1.04	1.11	1.25	0.66	0.76	0.81	0.81	0.89	1.00	0.99	1.03	1.21	0.63	0.75	0.82
FEX	1.02	1.14	1.31	1.26	1.74	1.61	0.81	0.77	1.02	1.02	1.20	1.28	1.38	1.57	1.99	0.75	0.77	0.90
OAS mean	0.94		1.27	1.05		1.38	0.88		1.17	0.94		1.27	1.03		1.44	0.83		1.14
OAS min	0.79		1.08	0.88		1.21	0.67		0.93	0.79		1.08	0.87		1.22	0.63		0.96
OAS max	1.12		1.48	1.49		1.78	0.98		1.31	1.11		1.47	1.34		1.69	0.94		1.29
	Real Subset Size			**Real Subset Size**			**Real Subset Size**			**Real Subset Size**			**Real Subset Size**			**Real Subset Size**		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
SS1	250	350		250	350		250	350		250	350		250	350		250	350	
SS10	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
SS1040	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
MAXMIN	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
FEX	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
OAS mean	237		578	239		583	235		571	236		565	238		572	233		560
OAS min	227		533	233		552	219		517	225		518	231		542	217		494
OAS max	244		610	248		624	241		596	243		601	248		616	239		587

Table 3: Average, minimum, and maximum of all performance measures over 50 artificial datasets of 2000 points. (*continued*)

B Results for artificial datasets of 5000 points

	Uniform Datasets									Non-Uniform Datasets								
	Average over datasets			Maximum over datasets			Minimum over datasets			Average over datasets			Maximum over datasets			Minimum over datasets		
	RMSE			RMSE			RMSE			RMSE			RMSE			RMSE		
RBF	0.07			0.08			0.07			0.08			0.08			0.07		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	0.17	0.14	0.12	0.19	0.17	0.15	0.14	0.12	0.11	0.18	0.15	0.13	0.22	0.18	0.16	0.15	0.13	0.11
SS1	0.10	0.08		0.12	0.11		0.09	0.07		0.10	0.08		0.12	0.09		0.09	0.07	
SS10	0.11	0.09	0.07	0.15	0.10	0.08	0.09	0.08	0.07	0.11	0.09	0.07	0.14	0.10	0.08	0.10	0.08	0.07
SS1040	0.10	0.08	0.07	0.12	0.10	0.07	0.09	0.07	0.06	0.10	0.08	0.07	0.14	0.10	0.08	0.09	0.08	0.06
MAXMIN	0.17	0.14	0.11	0.23	0.19	0.14	0.14	0.12	0.10	0.17	0.14	0.11	0.21	0.17	0.13	0.14	0.12	0.10
FEX	0.17	0.15	0.12	0.20	0.18	0.15	0.15	0.13	0.10	0.17	0.15	0.13	0.23	0.18	0.15	0.14	0.13	0.11
OAS mean	0.17		0.11	0.18		0.11	0.16		0.10	0.17		0.11	0.18		0.11	0.16		0.10
OAS min	0.15		0.10	0.16		0.11	0.14		0.09	0.15		0.10	0.17		0.11	0.14		0.09
OAS max	0.19		0.12	0.22		0.14	0.17		0.11	0.19		0.12	0.21		0.14	0.17		0.11
RBF	Maximum Error			Maximum Error			Maximum Error			Maximum Error			Maximum Error			Maximum Error		
	1.06			1.30			0.79			1.09			1.33			0.86		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	1.59	1.45	1.33	1.96	1.90	1.79	1.14	1.05	0.95	1.63	1.48	1.36	2.12	1.97	1.92	1.07	1.01	0.98
SS1	0.77	0.73		1.09	1.06		0.52	0.52		0.82	0.76		1.15	1.11		0.52	0.51	
SS10	0.77	0.74	0.69	1.10	1.01	0.96	0.55	0.51	0.42	0.82	0.77	0.75	1.17	1.08	1.12	0.55	0.49	0.48
SS1040	0.79	0.73	0.69	1.06	0.99	0.94	0.54	0.52	0.43	0.81	0.77	0.72	1.27	1.17	1.07	0.50	0.48	0.45
MAXMIN	1.58	1.39	1.15	2.29	2.08	1.39	1.09	1.11	0.84	1.47	1.28	1.10	1.90	1.72	1.42	1.11	1.00	0.77
FEX	1.61	1.50	1.29	1.95	2.01	1.76	1.23	0.92	0.74	1.59	1.47	1.33	1.92	1.93	1.70	1.07	1.00	0.96
OAS mean	1.59		1.15	1.75		1.36	1.48		0.97	1.58		1.14	1.68		1.36	1.42		1.01
OAS min	1.30		0.88	1.51		1.07	1.07		0.68	1.25		0.87	1.43		1.11	0.92		0.74
OAS max	1.92		1.46	2.24		1.78	1.70		1.24	1.87		1.45	2.19		1.70	1.60		1.21
RBF	Time Subset Selection			Time Subset Selection			Time Subset Selection			Time Subset Selection			Time Subset Selection			Time Subset Selection		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SS1	15.66	36.80		16.14	38.00		15.41	36.11		15.65	36.83		15.90	37.66		15.45	36.15	
SS10	1.54	3.63	9.70	1.57	3.71	9.99	1.51	3.57	9.46	1.54	3.63	9.72	1.59	3.80	9.87	1.52	3.56	9.54
SS1040	1.32	3.03	7.81	1.35	3.09	8.00	1.29	2.97	7.51	1.31	3.03	7.81	1.35	3.08	7.99	1.28	2.97	7.63
MAXMIN	1.20	1.61	2.22	1.23	1.64	2.25	1.16	1.57	2.18	1.20	1.60	2.22	1.23	1.63	2.26	1.16	1.57	2.18
FEX	0.45	0.52	0.72	0.62	0.72	1.11	0.30	0.38	0.44	0.57	0.77	1.06	0.72	1.03	1.47	0.38	0.53	0.77
OAS mean	0.01		0.03	0.01		0.03	0.01		0.03	0.01		0.03	0.01		0.03	0.01		0.03
OAS min	0.01		0.03	0.01		0.03	0.01		0.02	0.01		0.03	0.01		0.03	0.01		0.03
OAS max	0.01		0.03	0.01		0.04	0.01		0.03	0.01		0.03	0.01		0.04	0.01		0.03
RBF	Time Model Fitting			Time Model Fitting			Time Model Fitting			Time Model Fitting			Time Model Fitting			Time Model Fitting		
	1.86			2.14			1.79			1.84			2.08			1.77		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	0.10	0.21	0.51	0.11	0.22	0.53	0.09	0.19	0.49	0.10	0.21	0.51	0.11	0.22	0.53	0.09	0.20	0.48
SS1	0.10	0.20		0.10	0.22		0.09	0.19		0.10	0.20		0.10	0.21		0.09	0.19	
SS10	0.10	0.20	0.49	0.10	0.22	0.52	0.09	0.19	0.46	0.10	0.20	0.49	0.11	0.21	0.51	0.09	0.19	0.47
SS1040	0.08	0.16	0.37	0.08	0.17	0.40	0.07	0.15	0.35	0.08	0.16	0.37	0.08	0.17	0.39	0.07	0.15	0.35
MAXMIN	0.10	0.21	0.50	0.11	0.22	0.53	0.09	0.19	0.48	0.10	0.21	0.50	0.11	0.22	0.57	0.09	0.19	0.46
FEX	0.10	0.21	0.51	0.11	0.23	0.54	0.10	0.20	0.49	0.10	0.21	0.52	0.11	0.22	0.54	0.10	0.20	0.50
OAS mean	0.09		0.92	0.10		0.93	0.09		0.90	0.09		0.89	0.10		0.91	0.09		0.87
OAS min	0.09		0.82	0.09		0.87	0.08		0.77	0.09		0.79	0.09		0.83	0.08		0.75
OAS max	0.10		1.00	0.10		1.05	0.10		0.96	0.10		0.97	0.10		1.02	0.09		0.92

Table 4: Average, minimum, and maximum of all performance measures over 50 artificial datasets of 5000 points.

	Uniform Datasets								Non-Uniform Datasets							
	Average over datasets			Maximum over datasets			Minimum over datasets		Average over datasets			Maximum over datasets			Minimum over datasets	
	Condition number			Condition number			Condition number		Condition number			Condition number			Condition number	
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	1516	3242	5873	7140	25241	26979	217	718	1724	2028	4115	10185	10351	11534	28277	343	783	1584
SS1	185	408		681	1113		93	209		200	433		426	1054		102	193	
SS10	716	1128	1788	2704	4393	4515	200	457	713	1356	2158	3132	12587	16632	24440	348	511	965
SS1040	176	339	671	364	909	1381	62	192	374	199	404	960	371	890	2172	115	195	536
MAXMIN	129	283	602	422	877	1676	31	55	186	126	267	725	489	1114	3277	32	67	190
FEX	792	2161	4006	3857	15572	12351	129	380	582	1476	2292	5400	11319	10979	20195	187	239	812
OAS mean	462		4156	1689		6510	269		2497	497		4590	848		13719	266		2599
OAS min	144		1588	265		2714	59		637	163		1566	284		3007	71		710
OAS max	1194		9764	7357		23807	488		3804	1326		13113	3973		93526	566		4254
	Maximum Robustness			**Maximum Robustness**			**Maximum Robustness**			**Maximum Robustness**			**Maximum Robustness**			**Maximum Robustness**		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	5.96	8.39	10.54	15.50	21.88	34.12	2.03	2.13	2.97	8.51	11.28	15.67	30.62	35.52	61.64	2.33	3.03	3.20
SS1	2.16	2.67		4.24	6.38		1.32	1.55		2.33	3.06		4.87	8.11		1.40	1.54	
SS10	6.01	7.12	8.17	23.00	32.33	31.08	1.86	1.94	2.11	7.93	9.56	10.14	24.52	103.82	82.06	1.88	2.00	2.45
SS1040	1.62	1.80	1.97	2.50	2.61	2.93	1.22	1.45	1.61	1.68	1.90	2.16	2.61	2.50	3.74	1.34	1.46	1.75
MAXMIN	1.65	1.81	2.09	2.58	2.97	3.46	1.11	1.19	1.32	1.59	1.81	2.23	2.54	2.71	4.06	1.18	1.34	1.33
FEX	3.81	4.95	6.75	7.82	15.33	18.55	1.55	2.13	2.10	5.47	5.77	7.59	20.37	11.10	15.56	1.97	2.16	2.25
OAS mean	2.15		3.14	2.78		3.58	1.84		2.75	2.22		3.21	2.80		3.97	1.92		2.79
OAS min	1.62		2.39	2.05		2.65	1.32		1.71	1.65		2.41	1.99		2.82	1.34		2.00
OAS max	3.15		4.48	6.98		7.80	2.28		3.35	3.17		4.55	4.74		7.40	2.22		3.40
	Average Robustness			**Average Robustness**			**Average Robustness**			**Average Robustness**			**Average Robustness**			**Average Robustness**		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	1.11	1.35	1.66	2.35	2.63	3.35	0.51	0.91	1.09	1.22	1.56	1.98	2.37	2.50	3.46	0.69	0.87	1.09
SS1	0.42	0.52		0.68	0.74		0.28	0.32		0.42	0.52		0.68	0.77		0.27	0.36	
SS10	0.56	0.67	0.80	0.77	0.87	1.22	0.30	0.44	0.66	0.60	0.71	0.85	1.02	0.93	1.00	0.41	0.57	0.59
SS1040	0.66	0.71	0.79	0.73	0.78	0.84	0.56	0.66	0.71	0.66	0.71	0.81	0.75	0.78	0.90	0.55	0.63	0.72
MAXMIN	0.69	0.79	0.92	0.90	1.03	1.16	0.44	0.51	0.67	0.67	0.77	0.95	0.92	1.09	1.43	0.44	0.52	0.69
FEX	0.91	1.19	1.48	1.46	2.93	2.33	0.54	0.71	0.87	1.12	1.23	1.64	3.22	2.02	2.81	0.52	0.55	0.90
OAS mean	0.93		1.28	1.04		1.37	0.88		1.21	0.94		1.29	1.02		1.41	0.88		1.20
OAS min	0.78		1.12	0.89		1.26	0.62		0.95	0.80		1.11	0.88		1.25	0.67		1.01
OAS max	1.10		1.46	1.41		1.76	0.96		1.30	1.11		1.51	1.36		1.75	0.97		1.30
	Real Subset Size			**Real Subset Size**			**Real Subset Size**			**Real Subset Size**			**Real Subset Size**			**Real Subset Size**		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
SS1	250	350		250	350		250	350		250	350		250	350		250	350	
SS10	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
SS1040	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
MAXMIN	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
FEX	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
OAS mean	245		642	247		646	244		640	244		634	246		639	243		628
OAS min	240		614	243		627	236		600	238		604	243		617	233		587
OAS max	249		664	250		672	247		653	249		657	250		669	247		646

Table 4: Average, minimum, and maximum of all performance measures over 50 artificial datasets of 5000 points. (*continued*)

C Results for artificial datasets of 10000 points

Uniform Datasets

Average over datasets — RMSE

Subset size	250	350	500
RAND	0.17	0.15	0.13
SS1	0.10	0.08	
SS10	0.12	0.09	0.07
SS1040	0.12	0.09	0.07
MAXMIN	0.17	0.14	0.11
FEX	0.17	0.14	0.13
OAS mean	0.17		0.10
OAS min	0.15		0.09
OAS max	0.19		0.12

Maximum over datasets — RMSE

Subset size	250	350	500
RAND	0.21	0.18	0.15
SS1		0.15	0.09
SS10	0.15	0.10	0.08
SS1040	0.14	0.11	0.08
MAXMIN	0.20	0.17	0.15
FEX	0.20	0.16	0.14
OAS mean	0.18		0.11
OAS min	0.17		0.10
OAS max	0.21		0.13

Minimum over datasets — RMSE

Subset size	250	350	500
RAND	0.15	0.13	0.11
SS1	0.09	0.07	
SS10	0.10	0.08	0.06
SS1040	0.11	0.09	0.07
MAXMIN	0.14	0.12	0.10
FEX	0.14	0.13	0.10
OAS mean	0.16		0.10
OAS min	0.14		0.09
OAS max	0.18		0.11

Average over datasets — Maximum Error

Subset size	250	350	500
RAND	1.56	1.48	1.35
SS1	0.61	0.53	
SS10	0.64	0.57	0.51
SS1040	0.66	0.56	0.50
MAXMIN	1.61	1.39	1.11
FEX	1.60	1.40	1.32
OAS mean	1.60		1.11
OAS min	1.29		0.85
OAS max	1.93		1.42

Maximum over datasets — Maximum Error

Subset size	250	350	500
RAND	2.12	2.01	1.95
SS1	0.93	0.77	
SS10	0.92	0.87	0.76
SS1040	0.88	0.79	0.75
MAXMIN	2.37	1.92	1.79
FEX	1.99	1.87	1.86
OAS mean	1.75		1.25
OAS min	1.54		1.08
OAS max	2.33		1.70

Minimum over datasets — Maximum Error

Subset size	250	350	500
RAND	1.16	0.94	0.83
SS1	0.42	0.40	
SS10	0.49	0.42	0.36
SS1040	0.50	0.45	0.38
MAXMIN	1.12	1.06	0.78
FEX	1.12	1.01	0.91
OAS mean	1.45		0.96
OAS min	1.10		0.73
OAS max	1.71		1.15

Average over datasets — Time Subset Selection

Subset size	250	350	500
RAND	0.00	0.00	0.00
SS1	24.34	51.82	
SS10	2.41	5.16	12.15
SS1040	2.40	5.09	12.18
MAXMIN	2.75	3.57	4.79
FEX	0.83	0.95	1.18
OAS mean	0.02		0.05
OAS min	0.02		0.05
OAS max	0.02		0.05

Maximum over datasets — Time Subset Selection

Subset size	250	350	500
RAND	0.00	0.00	0.00
SS1	25.00	52.94	
SS10	2.74	5.43	12.54
SS1040	2.44	5.21	12.43
MAXMIN	2.80	3.65	4.87
FEX	1.16	1.40	1.92
OAS mean	0.02		0.05
OAS min	0.02		0.05
OAS max	0.03		0.07

Minimum over datasets — Time Subset Selection

Subset size	250	350	500
RAND	0.00	0.00	0.00
SS1	23.94	50.71	
SS10	2.37	5.06	11.71
SS1040	2.36	5.03	11.92
MAXMIN	2.61	3.42	4.64
FEX	0.59		0.93
OAS mean	0.02		0.05
OAS min	0.01		0.04
OAS max	0.02		0.05

Average over datasets — Time Model Fitting

Subset size	250	350	500
RAND	0.10	0.20	0.50
SS1	0.09	0.19	
SS10	0.10	0.19	0.46
SS1040	0.10	0.20	0.47
MAXMIN	0.10	0.20	0.49
FEX	0.10	0.21	0.50
OAS mean	0.09		0.96
OAS min	0.09		0.89
OAS max	0.10		1.01

Maximum over datasets — Time Model Fitting

Subset size	250	350	500
RAND	0.22	0.20	0.16
SS1	0.12	0.09	
SS10	0.13	0.11	0.09
SS1040	0.14	0.11	0.08
MAXMIN	0.21	0.18	0.14
FEX	0.22	0.19	0.15
OAS mean	0.18		0.11
OAS min	0.16		0.10
OAS max	0.21		0.13

Minimum over datasets — Time Model Fitting

Subset size	250	350	500
RAND	0.14	0.13	0.10
SS1	0.09	0.07	
SS10	0.10	0.08	0.06
SS1040	0.10	0.08	0.06
MAXMIN	0.14	0.12	0.10
FEX	0.15	0.13	0.11
OAS mean	0.16		0.10
OAS min	0.14		0.09
OAS max	0.17		0.11

Non-Uniform Datasets

Average over datasets — RMSE

Subset size	250	350	500
RAND	0.17	0.15	0.13
SS1	0.10	0.08	
SS10	0.12	0.09	0.07
SS1040	0.12	0.09	0.07
MAXMIN	0.17	0.14	0.11
FEX	0.17	0.14	0.13
OAS mean	0.17		0.10
OAS min	0.15		0.10
OAS max	0.19		0.12

Maximum over datasets — RMSE

Subset size	250	350	500
RAND	0.22	0.20	0.16
SS1	0.12	0.09	
SS10	0.13	0.11	0.09
SS1040	0.14	0.11	0.08
MAXMIN	0.21	0.18	0.14
FEX	0.22	0.19	0.15
OAS mean	0.18		0.11
OAS min	0.16		0.10
OAS max	0.21		0.13

Minimum over datasets — RMSE

Subset size	250	350	500
RAND	0.14	0.13	0.10
SS1	0.09	0.07	
SS10	0.10	0.08	0.06
SS1040	0.10	0.08	0.06
MAXMIN	0.14	0.12	0.10
FEX	0.15	0.13	0.11
OAS mean	0.16		0.10
OAS min	0.14		0.09
OAS max	0.17		0.11

Average over datasets — Maximum Error

Subset size	250	350	500
RAND	1.55	1.52	1.32
SS1	0.67	0.62	
SS10	0.69	0.62	0.58
SS1040	0.70	0.61	0.59
MAXMIN	1.58	1.34	1.11
FEX	1.64	1.45	1.38
OAS mean	1.58		1.13
OAS min	1.27		0.85
OAS max	1.89		1.42

Maximum over datasets — Maximum Error

Subset size	250	350	500
RAND	1.92	2.17	1.78
SS1	1.11	0.98	
SS10	1.02	0.97	0.89
SS1040	1.06	0.92	0.90
MAXMIN	1.95	1.84	1.50
FEX	1.92	1.86	1.74
OAS mean	1.75		1.24
OAS min	1.48		1.10
OAS max	2.15		1.72

Minimum over datasets — Maximum Error

Subset size	250	350	500
RAND	1.06	0.95	0.89
SS1	0.44	0.42	
SS10	0.51	0.42	0.39
SS1040	0.54	0.45	0.43
MAXMIN	1.10	0.76	0.76
FEX	1.25	1.00	0.87
OAS mean	1.40		1.00
OAS min	1.00		0.65
OAS max	1.63		1.25

Average over datasets — Time Subset Selection

Subset size	250	350	500
RAND	0.00	0.00	0.00
SS1	27.61	55.60	
SS10	2.53	5.27	12.51
SS1040	2.47	5.23	12.43
MAXMIN	2.77	3.59	4.82
FEX	1.36	1.77	2.40
OAS mean	0.02		0.05
OAS min	0.02		0.05
OAS max	0.03		0.06

Maximum over datasets — Time Subset Selection

Subset size	250	350	500
RAND	0.00	0.00	0.00
SS1	23.87	50.45	
SS10	2.37	5.03	11.88
SS1040	2.35	5.01	11.99
MAXMIN	2.61	3.42	4.64
FEX	0.86		1.61
OAS mean	0.02		0.05
OAS min	0.02		0.04
OAS max	0.02		0.05

Minimum over datasets — Time Subset Selection

Subset size	250	350	500
RAND	0.00	0.00	0.00
SS1	23.87	50.45	
SS10	2.37	5.03	11.88
SS1040	2.35	5.01	11.99
MAXMIN	2.61	3.42	4.64
FEX	0.86		1.61
OAS mean	0.02		0.05
OAS min	0.02		0.04
OAS max	0.02		0.05

Average over datasets — Time Model Fitting

Subset size	250	350	500
RAND	0.10	0.20	0.50
SS1	0.09	0.19	
SS10	0.10	0.19	0.46
SS1040	0.10	0.20	0.47
MAXMIN	0.10	0.20	0.49
FEX	0.10	0.21	0.50
OAS mean	0.09		0.96
OAS min	0.09		0.89
OAS max	0.10		1.01

Maximum over datasets — Time Model Fitting

Subset size	250	350	500
RAND	0.22	0.20	0.16
SS1	0.12	0.09	
SS10	0.13	0.11	0.09
SS1040	0.14	0.11	0.08
MAXMIN	0.21	0.18	0.14
FEX	0.22	0.19	0.15
OAS mean	0.18		0.11
OAS min	0.16		0.10
OAS max	0.21		0.13

Minimum over datasets — Time Model Fitting

Subset size	250	350	500
RAND	0.14	0.13	0.10
SS1	0.09	0.07	
SS10	0.10	0.08	0.06
SS1040	0.10	0.08	0.06
MAXMIN	0.14	0.12	0.10
FEX	0.15	0.13	0.11
OAS mean	0.16		0.10
OAS min	0.14		0.09
OAS max	0.17		0.11

Table 5: Average, minimum, and maximum of all performance measures over 50 artificial datasets of 10000 points.

	Uniform Datasets								Non-Uniform Datasets							
	Average over datasets			Maximum over datasets			Minimum over datasets		Average over datasets			Maximum over datasets			Minimum over datasets	
	Condition Number			Condition Number			Condition Number		Condition Number			Condition Number			Condition Number	
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	1553	2824	6239	6462	12486	31343	343	473	1179	2083	5611	11945	8990	61456	52456	459	1165	2194
SS1	175	355		341	823		77	191		195	419		362	983		79	186	
SS10	1025	1413	2194	2846	3330	4856	393	458	817	1730	2413	3759	6587	7510	11473	268	735	1337
SS1040	941	1638	2456	2162	3738	8439	462	599	969	1531	2399	4190	5206	9536	18334	420	705	1188
MAXMIN	109	221	685	302	855	2031	31	78	167	121	273	607	283	940	2142	32	72	150
FEX	1012	1886	5069	5271	5548	23544	234	533	732	1209	2937	4149	8944	14051	18497	226	469	1230
OAS mean	448		4296	891		6921	236		2990	447		4487	909		8439	257		2831
OAS min	141		1435	232		2653	55		614	145		1581	322		2517	83		599
OAS max	1226		10107	5348		33058	408		5288	1172		11156	3812		44761	435		4509
	Maximum Robustness			Maximum Robustness			Maximum Robustness			Maximum Robustness			Maximum Robustness			Maximum Robustness		
nsel	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
random	2.72	3.00	3.41	4.76	4.74	5.08	1.94	2.07	2.06	3.14	3.82	4.37	5.61	6.28	7.47	2.19	2.55	2.95
add 1	1.53	1.71		1.83	1.98		1.27	1.45		1.56	1.76		2.11	2.56		1.22	1.41	
add 10	3.08	3.10	3.23	4.49	5.68	5.57	1.92	2.28	2.40	3.51	3.63	3.84	7.30	6.74	6.90	1.89	2.13	2.42
add 1040	2.84	3.08	3.35	5.33	4.88	8.21	1.96	2.13	2.21	3.27	3.44	3.80	5.70	5.84	9.19	2.07	2.02	2.31
maxmin	1.35	1.36	1.53	1.77	1.62	1.92	1.08	1.19	1.30	1.32	1.40	1.50	1.68	1.71	2.01	1.06	1.16	1.23
exchange	2.30	2.41	2.86	4.21	3.32	4.59	1.72	1.91	1.88	2.33	2.74	2.96	3.67	3.88	4.60	1.53	1.93	2.23
oa mean	2.15		3.19	2.75		3.79	1.80		2.77	2.14		3.25	2.70		4.41	1.78		2.89
oa min	1.61		2.39	1.91		2.77	1.37		1.97	1.64		2.39	1.88		2.86	1.45		1.84
oa max	3.14		4.53	6.02		8.86	2.01		3.42	3.06		4.72	8.58		10.42	2.08		3.26
	Average Robustness			Average Robustness			Average Robustness			Average Robustness			Average Robustness			Average Robustness		
nsel	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
random	1.08	1.19	1.33	1.64	1.51	1.73	0.86	0.79	1.03	1.14	1.32	1.49	1.58	2.23	2.02	0.80	1.05	1.06
add 1	0.59	0.66		0.67	0.70		0.49	0.61		0.60	0.67		0.69	0.73		0.53	0.61	
add 10	0.76	0.80	0.87	0.92	0.89	0.98	0.63	0.74	0.74	0.79	0.85	0.92	1.01	1.06	1.02	0.61	0.75	0.78
add 1040	0.77	0.86	0.91	0.92	0.98	1.06	0.67	0.74	0.85	0.81	0.87	0.95	0.97	1.04	1.08	0.69	0.73	0.81
maxmin	0.80	0.86	0.96	0.92	1.00	1.10	0.68	0.77	0.84	0.80	0.88	0.95	0.91	1.02	1.13	0.67	0.74	0.82
exchange	1.04	1.14	1.30	1.47	1.47	1.87	0.80	0.93	0.97	1.05	1.21	1.29	1.59	1.76	1.76	0.78	0.87	1.02
oa mean	0.92		1.29	0.99		1.35	0.84		1.23	0.93		1.29	1.00		1.36	0.86		1.20
oa min	0.78		1.11	0.87		1.23	0.67		1.00	0.79		1.11	0.87		1.25	0.71		0.95
oa max	1.08		1.46	1.31		1.68	0.98		1.34	1.09		1.49	1.25		1.84	0.90		1.32
	Real Subset Size			Real Subset Size			Real Subset Size			Real Subset Size			Real Subset Size			Real Subset Size		
Subset size	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
RAND	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
SS1	250	350		250	350		250	350		250	350		250	350		250	350	
SS10	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
SS1040	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
MAXMIN	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
FEX	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500	250	350	500
OAS mean	248		666	249		669	247		663	247		660	248		663	246		656
OAS min	245		650	247		660	242		636	244		640	246		652	241		629
OAS max	250		678	250		683	249		670	250		675	250		680	248		668

Table 5: Average, minimum, and maximum of all performance measures over 50 artificial datasets of 10000 points. (*continued*)

D Results for HSCT dataset of 2487 points

	Results Srivastava	
Subset size	283	372
RMSPE	1.00	0.24
Avg. Perc. Err.	0.59	0.17
Max. Perc. Err.	6.12	2.02

	HSCT Dataset					
	RMSPE			**Condition number**		
Subset size	250	350	500	250	350	500
RAND	1.00	0.93	0.86	712731778	341942761	3072209097
OAS mean	1.30		1.11	112944986		1047914507
OAS min	1.05		0.86	2213		3344
OAS max	1.50		1.38	447474975		2377979099
FEX	1.72	1.73	1.73	2035776984	2328684961	1600933318
MAXMIN	1.31	1.14	1.22	68976	69945	55685
DELETION	1.70	1.83	1.46	29772	22407	69130
SS1	0.66	0.65		69777	197848	
SS10	0.60	1.28	0.68	1568232546	7678597938	9457605211
SS1040	0.58	0.44	0.36	162191	183002	108242
	Average Percentage Error			**Average Robustness**		
Subset size	250	350	500	250	350	500
RAND	0.59	0.52	0.52	43.08	15.11	27.80
OAS mean	0.85		0.69	19.06		63.38
OAS min	0.73		0.57	0.87		0.75
OAS max	0.95		0.81	62.81		117.07
FEX	1.19	1.17	1.21	177.70	182.17	142.40
MAXMIN	0.92	0.81	0.91	0.84	0.85	0.77
DELETION	1.22	1.33	1.08	1.61	1.66	1.06
SS1	0.53	0.51		1.50	1.35	
SS10	0.46	0.76	0.53	126.76	85.95	106.68
SS1040	0.44	0.35	0.27	1.17	1.03	0.91
	Maximum Percentage Error			**Real Subset Size**		
Subset size	250	350	500	250	350	500
RAND	9.37	8.96	7.65	250	350	500
OAS mean	5.77		5.67	226		505
OAS min	4.75		4.97	224		494
OAS max	7.11		6.82	230		515
FEX	8.59	9.19	8.45	250	350	500
MAXMIN	5.60	5.44	5.32	250	350	500
DELETION	7.55	7.76	6.51	250	350	500
SS1	2.02	2.34		250	350	
SS10	2.37	13.73	2.13	250	350	500
SS1040	2.36	1.76	2.50	250	350	500
	Time Subset Selection			**Time Model Fitting**		
Subset size	250	350	500	250	350	500
RAND	0.00	0.00	0.00	0.41	0.84	1.92
OAS mean	0.01		0.04	0.42		2.40
OAS min	0.01		0.04	0.23		2.23
OAS max	0.01		0.05	0.78		2.55
FEX	7.29	7.81	7.97	0.41	0.85	1.93
MAXMIN	0.12	0.12	0.13	0.38	0.56	1.21
DELETION	0.47	0.45	0.42	0.34	0.72	1.63
SS1	21.30	57.15		0.27	0.45	
SS10	2.64	6.08	16.72	0.23	0.43	0.97
SS1040	2.31	5.74	17.12	0.23	0.41	0.98

Table 6: Results for HSCT-dataset of 2487 points

E Kriging model

As we focus on Kriging, we summarize some theory according to Sacks et al. (1989). In Kriging, the output data $y(x)$ is treated as a realization of a random function $Y(x)$. This random function is divided into a regression part and a stochastic part:

$$Y(x) = \sum_{j=0}^{k} \beta_j f_j(x) + Z(x)$$

where $k+1$ is the number of regression functions including $f_0(x) = 1$. In this paper for the regression part, we use the linear functions $f_j(x) = x_j$ for $j = 1, \ldots, d$ where d is the number of dimensions. The stochastic part $Z(x)$ is assumed to have zero mean. Furthermore, the covariance between $Z(x)$ and $Z(w)$ is assumed to be of the form:

$$V(w, x) = \sigma^2 R(w, x)$$

where σ^2 is the constant process variance and $R(w, x)$ is the correlation between $Z(x)$ and $Z(w)$. To fit the Kriging model, we use a dataset with input data $X = [x^1, \ldots, x^n]$ and corresponding output data $y_X = [y(x^1), \ldots, y(x^n)]$. In Kriging, the vector y_X is assumed to be a realization of the stochastic vector $[Y(x^1), \ldots, Y(x^n)]$.

To predict the output value in a new point x, Kriging uses the Best Linear Unbiased Predictor (BLUP). This means that $\hat{y}(x)$, the predicted output value at point x, is given by:

$$\hat{y}(x) = c^T(x) y_X$$

where the Kriging weights $c(x)$ are determined such that they minimize:

$$MSE(\hat{y}(x)) = E(c^T(x) Y_X - Y(x))^2 \tag{1}$$

under the unbiasedness constraint:

$$E(c^T(x) Y_X) = E(Y(x)) \tag{2}$$

Let us now introduce the following notation:

$$\begin{aligned}
f(x) &= [f_0(x), f_1(x), \ldots, f_k(x)] = [1, x_1, \ldots, x_d]^T \\
F &= \begin{bmatrix} f^T(x^1) \\ \vdots \\ f^T(x^n) \end{bmatrix} \\
r(x) &= [R(x, x^1), \ldots, R(x, x^n)]^T \\
R &= \begin{bmatrix} r^T(x^1) \\ \vdots \\ r^T(x^n) \end{bmatrix} = \begin{bmatrix} R(x^1, x^1) & R(x^1, x^2) & \ldots & R(x^1, x^n) \\ R(x^2, x^1) & R(x^2, x^2) & \ldots & R(x^2, x^n) \\ \vdots & \vdots & \ddots & \vdots \\ R(x^n, x^1) & R(x^n, x^2) & \ldots & R(x^n, x^n) \end{bmatrix}
\end{aligned}$$

The matrix R is thus the correlation matrix containing the correlations between $Z(x^i)$ and $Z(x^j)$ for all $i, j \in \{1, \ldots, n\}$. By definition of the spatial correlation function $R(.)$, this matrix R is positive semi-definite and symmetric with ones on the diagonal.

Classical Kriging assumes that the Kriging weights $c(x)$ are independent of the output data. Therefore, we can rewrite the MSE in (1) as (Santner et al. 2003)

$$MSE(\hat{y}(x)) = \sigma^2 (1 + c^T(x) R c(x) - 2 c^T(x) r(x)) \tag{3}$$

and the constraint in (2) as:

$$F^T c(x) = f(x)$$

Using Lagrange multipliers $\lambda(x)$, we can minimize the MSE in (3) by solving the following system of equations:

$$\begin{bmatrix} 0 & F^T \\ F & R \end{bmatrix} \begin{bmatrix} \lambda(x) \\ c(x) \end{bmatrix} = \begin{bmatrix} f(x) \\ r(x) \end{bmatrix}$$

Solving this system gives the following expressions for $\lambda(x)$ and $c(x)$:

$$\begin{aligned}\lambda(x) &= (F^T R^{-1} F)^{-1}(F^T R^{-1} r(x) - f(x)) \\ c(x) &= R^{-1}(r(x) - F\lambda(x))\end{aligned}$$

We thus see that determining the prediction of the output at point x requires solving a linear system containing the $n \times n$ matrix R. If the size of the dataset, n, becomes very large this can be quite time consuming and we can thus save time by using less data.

Thus far, we have not specified the form of the correlation function $R(x, w)$. For the correlation function, there are a number of alternatives. We choose to use the Gaussian correlation function:

$$R^\theta(w, x) = \prod_{j=1}^{d} \exp\left(-\theta_j |w_j - x_j|^2\right)$$

as this is the most frequently used correlation function for Kriging (Jin et al. 2001). Furthermore, we assume that $Z(x)$ is a Gaussian process.

We now still have to determine β, σ and θ such that the Kriging model interpolates through the training data. To do this, we use the Maximum Likelihood Estimator (MLE). For β, this gives the generalized least-squares estimate:

$$\hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} y_X.$$

The MLE of σ^2 is given by:

$$\sigma^2 = \frac{1}{n}(y_X - F\hat{\beta})^T R^{-1}(y_X - F\hat{\beta}).$$

To determine the MLE of θ, we must solve the following minimization problem (Sacks et al. 1989):

$$\min_\theta |R|^{1/n} \hat{\sigma}^2$$

Unfortunately, we do not have an analytic expression for the $\hat{\theta}$ that solves this problem. We thus need some numerical optimization procedure to determine $\hat{\theta}$. As mentioned in Section 4, we use the DACE toolbox of Lophaven et al. (2002). Notice that this minimization problem also contains R^{-1} as it is part of the expression for $\hat{\theta}$. The numerical optimization procedure has to determine R for multiple values of θ in order to find the solution to the minimization problem. This can thus become very time-consuming if R is very large. As R is a $n \times n$ matrix, the size of the training set n directly affects the time-consumption of this step.

F Radial basis functions

Besides Kriging models, we can also use radial basis function (RBF) models to construct a model. RBF-models have been developed by Hardy (1971) to interpolate through a training set of multi-dimensional data. To approximate the output value in a point x, the distances between this point and the training points are used. Each distance value is used as the input of a radially symmetric function. A linear combination of the output values of this function forms the approximation of the output value in x. The simple RBF-model used in this paper to approximate y is:

$$\hat{y} = \sum_i \beta_i ||x - x_i||$$

where $||x - x_i||$ denotes the Euclidean distance between the training point x_i and the approximation point x. When we fill in all training points (x_i, y_i) for $i = 1, \ldots, n$, we obtain a set of linear equations. By solving this system, we can determine the coefficients β_i.

The advantage of RBF-models is that they have shown good fits to both stochastic and deterministic functions (Powell 1987) and that they require significantly less time to be fitted than Kriging models (Jin et al. 2001). Therefore, they are more suitable for larger datasets than Kriging. However, by combining subset selection and Kriging, we aim to make Kriging equally, or even better, suitable for large datasets. We therefore compare Kriging models fitted on a subset with radial basis functions fitted on the complete dataset. As some performance measures cannot be calculated for radial basis functions, we will only make a comparison of the accuracy and the time required to fit the model. For the Kriging models, this time also includes the time necessary to select the subset.